UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/708,027	02/03/2004	Yung-Chieh Lo	251812-1220	2026

24504 7590 07/17/2009

THOMAS, KAYDEN, HORSTEMEYER & RISLEY, LLP
600 GALLERIA PARKWAY, S.E.
STE 1500
ATLANTA, GA 30339-5994

EXAMINER	
HOUSHMAND, HOOMAN	

ART UNIT	PAPER NUMBER
2419	

MAIL DATE	DELIVERY MODE
07/17/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/708,027	LO ET AL.	
	Examiner	Art Unit	
	Hooman Houshmand	2419	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *14 May 2009*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-16 and 20-23* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-16, 20-23* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

1. Applicant's request for reconsideration of the finality of the rejection of the last Office action is persuasive and, therefore, the finality of that action is withdrawn.

2. Claims 1-16, 20-23 are pending.

Claim Rejections - 35 USC § 101

3. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

4. Claim(s) 1-16, 20-23 is/are rejected under 35 U.S.C. 101 as not falling within one of the four statutory categories of invention. While the claims recite a series of steps or acts to be performed, a statutory "process" under 35 U.S.C. 101 must (1) be tied to particular machine, or (2) transform underlying subject matter (such as an article or material) to a different state or thing. See page 10 of In Re Bilski 88 USPQ2d 1385. The instant claims are neither positively tied to a particular machine that accomplishes the claimed method steps nor transform underlying subject matter, and therefore do not qualify as a statutory process.

Claim Rejections - 35 USC § 112

5. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6. Claims 1-16, 20-23 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

7. The limitation (claim 1 lines 11-12) "*the remaining portion corresponds to a majority of the payload of the incoming packet*" is indefinite. Consider the case where the incoming packet has a payload of 1 byte. This payload will be transmitted with the first outgoing packet. There will be no remaining portion of payload to be transmitted with a second outgoing packet. Since there is no remaining portion of payload – how can *the remaining portion* of payload *correspond to a majority of the payload of the incoming packet*? In general, when the payload of the incoming packet is small enough – the entire payload will be transmitted with the first outgoing packet, and there will be no remaining portion of payload left to be transmitted with a second outgoing packet. In these cases there is no remaining portion of payload, and *the remaining portion* can not *correspond to a majority of the payload of the incoming packet*.

8. The limitation (claim 9 lines 9-10) "*the second fragment corresponds to a majority of the payload of the incoming packet*" is indefinite. Consider the case where the incoming packet has a payload of 1 byte. This payload will be transmitted with the first outgoing packet. There will be no *second fragment* of payload to be transmitted with a second outgoing packet. Since there is no *second fragment* of payload – how can *the second fragment* of payload *correspond to a majority of the payload of the incoming packet*? In general, when the payload of the incoming packet is small enough – the

entire payload will be transmitted with the first outgoing packet, and there will be no remaining *fragment* of payload left to be transmitted with a second outgoing packet. In these cases there is no remaining *fragment* of payload, and *the second fragment* can not *correspond to a majority of the payload of the incoming packet*.

9. The limitation (claim 20 lines 9-10) "*the size of the second outgoing packet is larger than that of the first outgoing packet*" is indefinite. Consider the case where the incoming packet has a payload of 1 byte. This payload will be transmitted with the first outgoing packet. There will be no second outgoing packet. Since there is no *second outgoing packet* – how can *the size of the second outgoing packet* be *larger than that of the first outgoing packet*? In general, when the payload of the incoming packet is small enough – the entire payload will be transmitted with the first outgoing packet, and there will be no *second outgoing packet*. In these cases there is no *second outgoing packet*, and *the size of the second outgoing packet* can not be *larger than that of the first outgoing packet*.

Claim Rejections - 35 USC § 103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

11. Claims 1-5, 8-13, 16, 20, 21, 22, 23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kitada (US 20030037163), in view of Applicant admitted prior art, and further in view of McCullough (US 20020010866 A1).

Claim 1. Kitada discloses *a method for fragmenting an incoming packet for transmission* (p11 [0185] perform fragment processing on IP packets in order to encapsulate the IP packets in accordance with PPPoE; FIG. 2C, [0188] when the IP data is encapsulated in accordance with PPPoE, the overhead of 8 bytes is included in the payload portion of 1,500 bytes, Therefore, the maximum transfer unit of the IP packet is reduced to 1,492 bytes. Hence, since the IP packet is transmitted in accordance with PPPoE, fragment processing is required).

Kitada does not teach the combination of these limitations: *a first outgoing packet and a second outgoing packet, storing a payload of the incoming packet in a plurality of storage units beginning in a first storage unit; transmitting the first outgoing packet being formed according to a predetermined portion of the payload stored in the first storage unit; and after transmitting the first outgoing packet, transmitting the second outgoing packet being formed according to a remaining portion of the payload stored in the storage units*.

In the same field of endeavor, Applicant admitted prior art discloses *a first outgoing packet* (Fig. 2 element 206) *and a second outgoing packet* (Fig. 2 element 210), *storing a payload of the incoming packet in a plurality of storage units beginning in a first storage unit* (Fig. 3 Buffers 1-12); *transmitting the first outgoing packet being*

formed according to a predetermined portion of the payload stored in the first storage unit (Fig. 3 up to the copy point; a first storage unit covering up to the copy point and a second storage unit covering the remaining bits); *and after transmitting the first outgoing packet, transmitting the second outgoing packet being formed according to a remaining portion of the payload stored in the storage units* (Fig. 2 element 210 Fragment 2 element 208).

It would have been obvious to a person having ordinary skill in the art, at the time that the invention was made, to combine the teachings of the Applicant's admitted prior art with Kitada to implement (e.g. in hardware) Kitada's IP data encapsulation in accordance with PPPoE protocol.

Kitada discloses *a method for fragmenting an incoming packet for transmission* (p11 [0185] perform fragment processing on IP packets in order to encapsulate the IP packets in accordance with PPPoE; FIG. 2C, [0188] when the IP data is encapsulated in accordance with PPPoE, the overhead of 8 bytes is included in the payload portion of 1,500 bytes, Therefore, the maximum transfer unit of the IP packet is reduced to 1,492 bytes. Hence, since the IP packet is transmitted in accordance with PPPoE, fragment processing is required).

It would have been obvious to a person having ordinary skill in the art, at the time that the invention was made, that Kitada's disclosure that fragment processing is required means that the payload of the incoming packet is fragmented into portions for transmission. Given an option of two portions, it would have been obvious to a person

having ordinary skill in the art that these portions are either equal or one is larger than the other one. Hence when transmitting these portions either the packets are of equal size or one packet is larger than the other packet.

However, Kitada does not disclose that *the remaining portion corresponds to a majority of the payload of the incoming packet*.

In the same field of endeavor, McCullough discloses ([0078] The bundle manager fragments a packet by comparing the size of the packet with the transmission unit MTU, which for underlying PPP links is 1500 bytes, [0079] In the gateway device, the fragment size is set at configuration time to 50% of the PPP MTU, this setting can be overridden, the upper limit is the full MTU. when there is a large transfer of data between peers, the bundle manager distributes a 1500 byte fragment on each available link in a round-robin fashion, or numerous small transfers are interleaved with fewer large transfers. fragment size is tuned for different circumstances to achieve the best aggregate throughput.) *the remaining portion corresponds to a majority of the payload of the incoming packet*.

It would have been obvious to a person having ordinary skill in the art, at the time that the invention was made, to combine the teachings of the McCullough with Kitada, thus modifying Kitada to have a small transfer followed by a large transfer, to avoid network congestion and adaptively achieve the best aggregate throughput. E.g., when the network is congested – it is advantages to send a small size packet first, this will keep the connection alive and have the least impact on bandwidth resources. Another

motivation is a fast start – by sending out a small size packet first, the receiver would receive some data faster and would start the communication earlier.

Claims 2, 21. Kitada further teaches *the first and second outgoing packets are Point-to-Point Protocol over Ethernet frames* ([0185] perform fragment processing in order to encapsulate in accordance with PPPoE; FIG. 2C).

Claims 3, 23. Kitada does not teach the combination of these limitations: *generating a first outgoing sub-header according to a header of the incoming packet and the predetermined portion of the payload stored in the first storage unit; generating a second outgoing sub-header according to the header of the incoming packet or the first outgoing sub-header, and the remaining portion of the payload; including the first outgoing sub-header and the predetermined portion of the payload stored in the first storage unit in the first outgoing packet; and including the second outgoing sub-header and the remaining portion of the payload stored in the storage units in the second outgoing packet*.

In the same field of endeavor, Applicant admitted prior art discloses *generating a first outgoing sub-header according to a header of the incoming packet and the predetermined portion of the payload stored in the first storage unit* (Fig. 2 element 206, element 204– as modified as the result of the combination in the parent claim); *generating a second outgoing sub-header according to the header of the incoming packet or the first outgoing sub-header, and the remaining portion of the payload* (Fig. 2

element 210, element 208– as modified as the result of the combination in the parent claim); *including the first outgoing sub-header and the predetermined portion of the payload stored in the first storage unit in the first outgoing packet* (Fig. 3 up to the copy point– as modified as the result of the combination in the parent claim); *and including the second outgoing sub-header and the remaining portion of the payload stored in the storage units in the second outgoing packet* (Fig. 3 after the Copy point– as modified as the result of the combination in the parent claim).

It would have been obvious to a person having ordinary skill in the art, at the time that the invention was made, to combine the teachings of the Applicant's admitted prior art with Kitada to implement Kitada's IP data encapsulation in accordance with PPPoE protocol.

Claims 4, 22. Kitada further teaches *the incoming packet is an Internet Protocol packet and the header of the incoming packet is the IP header of the incoming IP packet* ([0185] perform fragment processing on IP packets).

Claim 5. Kitada does not teach the combination of these limitations: *the first outgoing sub-header is a first IP header corresponding to the predetermined portion of the payload stored in the first storage unit and the incoming IP header, and the second outgoing sub-header is a second IP header corresponding to the remaining portion of the payload, and the incoming IP header or the first outgoing sub-header*.

In the same field of endeavor, Applicant admitted prior art discloses *the first outgoing sub-header is a first IP header corresponding to the predetermined portion of the payload stored in the first storage unit* (Fig. 3 up to the copy point– as modified as the result of the combination in the parent claim) *and the incoming IP header* (Fig. 2 element 206, element 204), *and the second outgoing sub-header is a second IP header corresponding to the remaining portion of the payload* (Fig. 3 after the Copy point– as modified as the result of the combination in the parent claim), *and the incoming IP header or the first outgoing sub-header* (Fig. 2 element 210, element 208– as modified as the result of the combination in the parent claim).

It would have been obvious to a person having ordinary skill in the art, at the time that the invention was made, to combine the teachings of the Applicant's admitted prior art with Kitada to implement Kitada's IP data encapsulation in accordance with PPPoE protocol.

Claim 8. Kitada does not teach the combination of these limitations: *the first outgoing sub-header and the first fragment are included as a first outgoing payload of the first outgoing packet, and the second outgoing sub-header and the second fragment are included as a second outgoing payload of the second outgoing packet*.

In the same field of endeavor, Applicant admitted prior art discloses *the first outgoing sub-header and the first fragment are included as a first outgoing payload of the first outgoing packet* (Fig. 2 element 206– as modified as the result of the combination in the parent claim), *and the second outgoing sub-header and the second*

fragment are included as a second outgoing payload of the second outgoing packet (Fig. 2 element 210– as modified as the result of the combination in the parent claim).

It would have been obvious to a person having ordinary skill in the art, at the time that the invention was made, to combine the teachings of the Applicant's admitted prior art with Kitada to implement Kitada's IP data encapsulation in accordance with PPPoE protocol.

Claim 9. Kitada discloses *a method for fragmenting an incoming packet for inclusion* ([0185] perform fragment processing on IP packets in order to encapsulate the IP packets in accordance with PPPoE; FIG. 2C, [0188] when the IP data is encapsulated in accordance with PPPoE, the overhead of 8 bytes is included in the payload portion of 1,500 bytes, Therefore, the maximum transfer unit of the IP packet is reduced to 1,492 bytes. Hence, since the IP packet is transmitted in accordance with PPPoE, fragment processing is required)

Kitada does not teach the combination of these limitations: *a first outgoing packet and a second outgoing packet, storing a payload of the incoming packet as a first fragment and a second fragment in a plurality of storage units; including the first fragment in the first outgoing packet; and after including the first fragment in the first outgoing packet, including the second fragment in the second outgoing packet*.

Applicant admitted prior art discloses *a first outgoing packet* (Fig. 2 element 206) *and a second outgoing packet* (Fig. 2 element 210), *storing a payload of the incoming packet as a first fragment and a second fragment in a plurality of storage units* (Fig. 3

Buffers 1-12)*; including the first fragment in the first outgoing packet, and after including the first fragment in the first outgoing packet, including the second fragment in the second outgoing packet* (Fig. 2 element 210 Fragment 2 element 208).

It would have been obvious to a person having ordinary skill in the art, at the time that the invention was made, to combine the teachings of the Applicant's admitted prior art with Kitada to implement Kitada's IP data encapsulation in accordance with PPPoE protocol.

Kitada discloses *a method for fragmenting an incoming packet for transmission* (p11 [0185] perform fragment processing on IP packets in order to encapsulate the IP packets in accordance with PPPoE; FIG. 2C, [0188] when the IP data is encapsulated in accordance with PPPoE, the overhead of 8 bytes is included in the payload portion of 1,500 bytes, Therefore, the maximum transfer unit of the IP packet is reduced to 1,492 bytes. Hence, since the IP packet is transmitted in accordance with PPPoE, fragment processing is required).

It would have been obvious to a person having ordinary skill in the art, at the time that the invention was made, that Kitada's disclosure that fragment processing is required means that the payload of the incoming packet is fragmented into portions for transmission. Given an option of two portions, it would have been obvious to a person having ordinary skill in the art that these portions are either equal or one is larger than the other one. Hence when transmitting these portions either the packets are equal size or one packet is larger than the other packet.

However, Kitada does not disclose that *the second fragment corresponds to a majority of the payload of the incoming packet*.

In the same field of endeavor, McCullough discloses ([0078] The bundle manager fragments a packet by comparing the size of the packet with the transmission unit MTU, which for underlying PPP links is 1500 bytes, [0079] In the gateway device, the fragment size is set at configuration time to 50% of the PPP MTU, this setting can be overridden, the upper limit is the full MTU. when there is a large transfer of data between peers, the bundle manager distributes a 1500 byte fragment on each available link in a round-robin fashion, or numerous small transfers are interleaved with fewer large transfers. fragment size is tuned for different circumstances to achieve the best aggregate throughput) *the second fragment corresponds to a majority of the payload of the incoming packet*.

It would have been obvious to a person having ordinary skill in the art, at the time that the invention was made, to combine the teachings of the McCullough with Kitada, thus modifying Kitada to have a small transfer followed by a large transfer, to avoid network congestion and adaptively achieve the best aggregate throughput. E.g., when the network is congested – it is advantages to send a small size packet first, this will keep the connection alive and have the least impact on bandwidth resources. Another motivation is a fast start – by sending out a small size packet first, the receiver would receive some data faster and would start the communication earlier.

Claim 10. Kitada further teaches *the incoming packet is an Internet Protocol packet received in an Ethernet frame and the first and second outgoing packets are Point-to-Point Protocol over Ethernet frames* ([0185] perform fragment processing on IP packets in order to encapsulate the IP packets in accordance with PPPoE; FIG. 2C).

Claim 11. Kitada does not teach the combination of these limitations: *generating a first outgoing sub-header and a second outgoing sub-header according to the first fragment, the second fragment, and a header of the incoming packet; including the first outgoing sub-header in the first outgoing packet; and including the second outgoing sub-header in the second outgoing packet*.

In the same field of endeavor, Applicant admitted prior art discloses *generating a first outgoing sub-header and a second outgoing sub-header according to the first fragment, the second fragment, and a header of the incoming packet; including the first outgoing sub-header in the first outgoing packet; and including the second outgoing sub-header in the second outgoing packet* (Fig. 2 elements 200, 206, 210).

It would have been obvious to a person having ordinary skill in the art, at the time that the invention was made, to combine the teachings of the Applicant's admitted prior art with Kitada to implement Kitada's IP data encapsulation in accordance with PPPoE protocol.

Claim 12. Kitada further teaches *the incoming packet is an incoming Internet Protocol packet and the header of the incoming packet is the IP header of the incoming IP packet* ([0185] perform fragment processing on IP packets).

Claim 13. Kitada does not teach the combination of these limitations: *the first outgoing sub-header is a first outgoing IP header generated corresponding to the first fragment and the IP header of the incoming IP packet, and the second outgoing sub-header is a second outgoing IP header generated corresponding to the second fragment, and the IP header of the incoming IP packet or the first outgoing sub-header*.

In the same field of endeavor, Applicant admitted prior art discloses *the first outgoing sub-header is a first outgoing IP header generated corresponding to the first fragment and the IP header of the incoming IP packet, and the second outgoing sub-header is a second outgoing IP header generated corresponding to the second fragment, and the IP header of the incoming IP packet or the first outgoing sub-header* (Fig. 2 elements 200, 206, 210).

It would have been obvious to a person having ordinary skill in the art, at the time that the invention was made, to combine the teachings of the Applicant's admitted prior art with Kitada to implement Kitada's IP data encapsulation in accordance with PPPoE protocol.

Claim 16. Kitada does not teach the combination of these limitations: *the first outgoing sub-header and the first fragment are included in a payload of the first outgoing packet,*

and the second outgoing sub-header and the second fragment are included in a payload of the second outgoing packet.

In the same field of endeavor, Applicant admitted prior art discloses *the first outgoing sub-header and the first fragment are included in a payload of the first outgoing packet* (Fig. 2 element 206– as modified as the result of the combination in the parent claim), *and the second outgoing sub-header and the second fragment are included in a payload of the second outgoing packet* (Fig. 2 element 210– as modified as the result of the combination in the parent claim).

It would have been obvious to a person having ordinary skill in the art, at the time that the invention was made, to combine the teachings of the Applicant's admitted prior art with Kitada to implement Kitada's IP data encapsulation in accordance with PPPoE protocol.

Claim 20. Kitada discloses *a method for fragmenting an incoming packet for transmission as first and second outgoing packets* (p11 [0185] perform fragment processing on IP packets in order to encapsulate the IP packets in accordance with PPPoE; FIG. 2C, [0188] when the IP data is encapsulated in accordance with PPPoE, the overhead of 8 bytes is included in the payload portion of 1,500 bytes, Therefore, the maximum transfer unit of the IP packet is reduced to 1,492 bytes. Hence, since the IP packet is transmitted in accordance with PPPoE, fragment processing is required).

Kitada does not teach the combination of these limitations:

storing payload of the incoming packet in a storage unit;

transmitting the first outgoing packet being formed according to a predetermined portion of the payload stored in the storage unit; and

after transmitting the first outgoing packet, transmitting the second outgoing packet being formed according to a remaining portion of the payload stored in the storage unit.

In the same field of endeavor, Applicant admitted prior art discloses *storing payload of the incoming packet in a storage unit* (Fig. 3 Buffers 1-12)*; transmitting the first outgoing packet* (Fig. 2 element 206) *being formed according to a predetermined portion of the payload stored in the storage unit* (Fig. 3 up to the copy point; a first storage unit covering up to the copy point and a second storage unit covering the remaining bits)*; and*

after transmitting the first outgoing packet, transmitting the second outgoing packet (Fig. 2 element 210) *being formed according to a remaining portion of the payload stored in the storage unit* (Fig. 2 element 210 Fragment 2 element 208).

It would have been obvious to a person having ordinary skill in the art, at the time that the invention was made, to combine the teachings of the Applicant's admitted prior art with Kitada to implement Kitada's IP data encapsulation in accordance with PPPoE protocol.

Kitada discloses *a method for fragmenting an incoming packet for transmission* (p11 [0185] perform fragment processing on IP packets in order to encapsulate the IP packets in accordance with PPPoE; FIG. 2C, [0188] when the IP data is encapsulated in accordance with PPPoE, the overhead of 8 bytes is included in the payload portion of 1,500 bytes, Therefore, the maximum transfer unit of the IP packet is reduced to 1,492 bytes. Hence, since the IP packet is transmitted in accordance with PPPoE, fragment processing is required).

It would have been obvious to a person having ordinary skill in the art, at the time that the invention was made, that Kitada's disclosure that fragment processing is required means that the payload of the incoming packet is fragmented into portions for transmission. Given an option of two portions, it would have been obvious to a person having ordinary skill in the art that these portions are either equal or one is larger than the other one. Hence when transmitting these portions either the packets are of equal size or one packet is larger than the other packet.

However, Kitada does not disclose that *the size of the second outgoing packet is larger than that of the first outgoing packet*.

In the same field of endeavor, McCullough discloses ([0078] The bundle manager fragments a packet by comparing the size of the packet with the transmission unit MTU, which for underlying PPP links is 1500 bytes, [0079] In the gateway device, the fragment size is set at configuration time to 50% of the PPP MTU, this setting can be overridden, the upper limit is the full MTU. when there is a large transfer of data between peers, the bundle manager distributes a 1500 byte fragment on each available

link in a round-robin fashion, or numerous small transfers are interleaved with fewer large transfers. fragment size is tuned for different circumstances to achieve the best aggregate throughput.) *the size of the second outgoing packet is larger than that of the first outgoing packet*.

It would have been obvious to a person having ordinary skill in the art, at the time that the invention was made, to combine the teachings of the McCullough with Kitada, thus modifying Kitada to have a small transfer followed by a large transfer, to avoid network congestion and adaptively achieve the best aggregate throughput. E.g., when the network is congested – it is advantages to send a small size packet first, this will keep the connection alive and have the least impact on bandwidth resources. Another motivation is a fast start – by sending out a small size packet first, the receiver would receive some data faster and would start the communication earlier.

12. Claims 6-7, 14-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kitada, in view of Applicant admitted prior art, further in view of McCullough as applied to claims 1, 9 above, and further in view of Kitamura (US 20030065799).

Claim 6. Kitada does not teach the combination of these limitations: *generating the first outgoing sub-header comprises modifying the MF, Offset, Length, and Checksum fields of the incoming IP header according to the predetermined portion of the payload stored in the first storage unit*.

In the same field of endeavor, Kitamura discloses *generating the first outgoing sub-header comprises modifying the MF, Offset, Length, and Checksum fields of the incoming IP header according to the predetermined portion of the payload stored in the first storage unit* (Fig. 8, p6 [0118] expanded PPPoE portion, an upper layer identifier, a version, a type, a TOS, a data length, an identifier, a flag, a fragment offset, a TTL, an upper layer identifier, a header checksum, a source IP address, a destination IP address are written).

It would have been obvious to a person having ordinary skill in the art, at the time that the invention was made, to combine the teachings of Kitamura with Kitada to follow the same arrangement as is customary in packet communications when one packet is fragmented into multiple packets.

Claim 7. Kitada does not teach the combination of these limitations: *generating the second outgoing sub-header comprises modifying the MF, Offset, Length, and Checksum fields of the incoming packet IP header or the first outgoing sub-header according to the remaining portion of the payload stored in the storage units*.

In the same field of endeavor, Kitamura discloses *generating the second outgoing sub-header comprises modifying the MF, Offset, Length, and Checksum fields of the incoming packet IP header or the first outgoing sub-header according to the remaining portion of the payload stored in the storage units* (Fig. 8, [0118] expanded PPPoE portion, an upper layer identifier, a version, a type, a TOS, a data length, an

identifier, a flag, a fragment offset, a TTL, an upper layer identifier, a header checksum, a source IP address, a destination IP address are written).

It would have been obvious to a person having ordinary skill in the art, at the time that the invention was made, to combine the teachings of Kitamura with Kitada to follow the same arrangement as is customary in packet communications when one packet is fragmented into multiple packets.

Claim 14. Kitada does not teach the combination of these limitations: *generating the first outgoing sub-header modifies the MF, Offset, Length, and checksum fields of the incoming packet IP header according to the first fragment*.

In the same field of endeavor, Kitamura discloses *generating the first outgoing sub-header modifies the MF, Offset, Length, and checksum fields of the incoming packet IP header according to the first fragment* (Fig. 8, [0118] expanded PPPoE portion, an upper layer identifier, a version, a type, a TOS, a data length, an identifier, a flag, a fragment offset, a TTL, an upper layer identifier, a header checksum, a source IP address, a destination IP address are written).

It would have been obvious to a person having ordinary skill in the art, at the time that the invention was made, to combine the teachings of Kitamura with Kitada to follow the same arrangement as is customary in packet communications when one packet is fragmented into multiple packets.

Claim 15. Kitada does not teach the combination of these limitations: *generating the second outgoing sub-header modifies the MF, offset, length, and checksum fields of the incoming packet IP header or the first outgoing sub-header according to the second fragment*.

In the same field of endeavor, Kitamura discloses *generating the second outgoing sub-header modifies the MF, offset, length, and checksum fields of the incoming packet IP header or the first outgoing sub-header according to the second fragment* (Fig. 8, [0118] expanded PPPoE portion, an upper layer identifier, a version, a type, a TOS, a data length, an identifier, a flag, a fragment offset, a TTL, an upper layer identifier, a header checksum, a source IP address, a destination IP address are written).

It would have been obvious to a person having ordinary skill in the art, at the time that the invention was made, to combine the teachings of Kitamura with Kitada to follow the same arrangement as is customary in packet communications when one packet is fragmented into multiple packets.

13. Claims 1-5, 8-13, 16, 20, 21, 22, 23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Applicant admitted prior art (AAPA), in view of McCullough.

Claim 1. AAPA discloses *a method for fragmenting an incoming packet for transmission* as *a first outgoing packet* (Fig. 2 element 206) *and a second outgoing packet* (Fig. 2 element 210), *storing a payload of the incoming packet in a plurality of storage units*

beginning in a first storage unit (Fig. 3 Buffers 1-12); *transmitting the first outgoing packet being formed according to a predetermined portion of the payload stored in the first storage unit* (Fig. 3 up to the copy point; a first storage unit covering up to the copy point and a second storage unit covering the remaining bits); *and after transmitting the first outgoing packet, transmitting the second outgoing packet being formed according to a remaining portion of the payload stored in the storage units* (Fig. 2 element 210 Fragment 2 element 208).

AAPA does not disclose that *the remaining portion corresponds to a majority of the payload of the incoming packet*.

In the same field of endeavor, McCullough discloses ([0078] The bundle manager fragments a packet by comparing the size of the packet with the transmission unit MTU, which for underlying PPP links is 1500 bytes, [0079] In the gateway device, the fragment size is set at configuration time to 50% of the PPP MTU, this setting can be overridden, the upper limit is the full MTU. when there is a large transfer of data between peers, the bundle manager distributes a 1500 byte fragment on each available link in a round-robin fashion, or numerous small transfers are interleaved with fewer large transfers. fragment size is tuned for different circumstances to achieve the best aggregate throughput.) *the remaining portion corresponds to a majority of the payload of the incoming packet*.

It would have been obvious to a person having ordinary skill in the art, at the time that the invention was made, to combine the teachings of the McCullough with AAPA,

thus modifying AAPA to have a small transfer followed by a large transfer, to avoid network congestion and adaptively achieve the best aggregate throughput. E.g., when the network is congested – it is advantages to send a small size packet first, this will keep the connection alive and have the least impact on bandwidth resources. Another motivation is a fast start – by sending out a small size packet first, the receiver would receive some data faster and would start the communication earlier.

Claims 2, 21. AAPA further teaches *the first and second outgoing packets are Point-to-Point Protocol over Ethernet frames* (FIG. 2).

Claims 3, 23. AAPA further discloses *generating a first outgoing sub-header according to a header of the incoming packet and the predetermined portion of the payload stored in the first storage unit* (Fig. 2 element 206, element 204– as modified as the result of the combination in the parent claim); *generating a second outgoing sub-header according to the header of the incoming packet or the first outgoing sub-header, and the remaining portion of the payload* (Fig. 2 element 210, element 208– as modified as the result of the combination in the parent claim); *including the first outgoing sub-header and the predetermined portion of the payload stored in the first storage unit in the first outgoing packet* (Fig. 3 up to the copy point – AAPA had been modified in the parent claims 1 and 20 so that the copy point now represents a small portion of the incoming packet. Fig. 3 is the replacement Fig. 3 Prior art which was filed on 07/31/2008); *and including the second outgoing sub-header and the remaining portion of the payload*

stored in the storage units in the second outgoing packet (Fig. 3 after the Copy point– as modified as the result of the combination in the parent claim).

Claims 4, 22. AAPA further teaches *the incoming packet is an Internet Protocol packet and the header of the incoming packet is the IP header of the incoming IP packet* (Fig. 2).

Claim 5. AAPA further discloses *the first outgoing sub-header is a first IP header corresponding to the predetermined portion of the payload stored in the first storage unit* (Fig. 3 up to the copy point – as modified as the result of the combination in the parent claim) *and the incoming IP header* (Fig. 2 element 206, element 204), *and the second outgoing sub-header is a second IP header corresponding to the remaining portion of the payload* (Fig. 3 after the Copy point– as modified as the result of the combination in the parent claim), *and the incoming IP header or the first outgoing sub-header* (Fig. 2 element 210, element 208).

Claim 8. AAPA further discloses *the first outgoing sub-header and the first fragment are included as a first outgoing payload of the first outgoing packet* (Fig. 2 element 206– as modified as the result of the combination in the parent claim), *and the second outgoing sub-header and the second fragment are included as a second outgoing payload of the second outgoing packet* (Fig. 2 element 210– as modified as the result of the combination in the parent claim).

Claim 9. AAPA discloses *a method for fragmenting an incoming packet for inclusion in a first outgoing packet* (Fig. 2 element 206) *and a second outgoing packet* (Fig. 2 element 210), *storing a payload of the incoming packet as a first fragment and a second fragment in a plurality of storage units* (Fig. 3 Buffers 1-12)*; including the first fragment in the first outgoing packet, and after including the first fragment in the first outgoing packet, including the second fragment in the second outgoing packet* (Fig. 2 element 210 Fragment 2 element 208).

AAPA does not disclose that *the second fragment corresponds to a majority of the payload of the incoming packet*.

In the same field of endeavor, McCullough discloses ([0078] The bundle manager fragments a packet by comparing the size of the packet with the transmission unit MTU, which for underlying PPP links is 1500 bytes, [0079] In the gateway device, the fragment size is set at configuration time to 50% of the PPP MTU, this setting can be overridden, the upper limit is the full MTU. when there is a large transfer of data between peers, the bundle manager distributes a 1500 byte fragment on each available link in a round-robin fashion, or numerous small transfers are interleaved with fewer large transfers. fragment size is tuned for different circumstances to achieve the best aggregate throughput) *the second fragment corresponds to a majority of the payload of the incoming packet*.

It would have been obvious to a person having ordinary skill in the art, at the time that the invention was made, to combine the teachings of the McCullough with AAPA, thus modifying AAPA to have a small transfer followed by a large transfer, to avoid network congestion and adaptively achieve the best aggregate throughput. E.g., when the network is congested – it is advantages to send a small size packet first, this will keep the connection alive and have the least impact on bandwidth resources. Another motivation is a fast start – by sending out a small size packet first, the receiver would receive some data faster and would start the communication earlier.

Claim 10. AAPA further teaches *the incoming packet is an Internet Protocol packet received in an Ethernet frame and the first and second outgoing packets are Point-to-Point Protocol over Ethernet frames* (FIG. 2).

Claim 11. AAPA discloses *generating a first outgoing sub-header and a second outgoing sub-header according to the first fragment, the second fragment, and a header of the incoming packet; including the first outgoing sub-header in the first outgoing packet; and including the second outgoing sub-header in the second outgoing packet* (Fig. 2 elements 200, 206, 210).

Claim 12. AAPA further teaches *the incoming packet is an incoming Internet Protocol packet and the header of the incoming packet is the IP header of the incoming IP packet* (FIG. 2).

Claim 13. AAPA discloses *the first outgoing sub-header is a first outgoing IP header generated corresponding to the first fragment and the IP header of the incoming IP packet, and the second outgoing sub-header is a second outgoing IP header generated corresponding to the second fragment, and the IP header of the incoming IP packet or the first outgoing sub-header* (Fig. 2 elements 200, 206, 210).

Claim 16. AAPA discloses *the first outgoing sub-header and the first fragment are included in a payload of the first outgoing packet* (Fig. 2 element 206– as modified as the result of the combination in the parent claim), *and the second outgoing sub-header and the second fragment are included in a payload of the second outgoing packet* (Fig. 2 element 210– as modified as the result of the combination in the parent claim).

Claim 20. AAPA *a method for fragmenting an incoming packet for transmission as first and second outgoing packets, storing payload of the incoming packet in a storage unit* (Fig. 3 Buffers 1-12)*;*
transmitting the first outgoing packet (Fig. 2 element 206) *being formed according to a predetermined portion of the payload stored in the storage unit* (Fig. 3 up to the copy point; a first storage unit covering up to the copy point and a second storage unit covering the remaining bits)*; and*

after transmitting the first outgoing packet, transmitting the second outgoing packet (Fig. 2 element 210) *being formed according to a remaining portion of the payload stored in the storage unit* (Fig. 2 element 210 Fragment 2 element 208).

AAPA does not disclose that *the size of the second outgoing packet is larger than that of the first outgoing packet*.

In the same field of endeavor, McCullough discloses ([0078] The bundle manager fragments a packet by comparing the size of the packet with the transmission unit MTU, which for underlying PPP links is 1500 bytes, [0079] In the gateway device, the fragment size is set at configuration time to 50% of the PPP MTU, this setting can be overridden, the upper limit is the full MTU. when there is a large transfer of data between peers, the bundle manager distributes a 1500 byte fragment on each available link in a round-robin fashion, or numerous small transfers are interleaved with fewer large transfers. fragment size is tuned for different circumstances to achieve the best aggregate throughput.) *the size of the second outgoing packet is larger than that of the first outgoing packet*.

It would have been obvious to a person having ordinary skill in the art, at the time that the invention was made, to combine the teachings of the McCullough with AAPA, thus modifying AAPA to have a small transfer followed by a large transfer, to avoid network congestion and adaptively achieve the best aggregate throughput. E.g., when the network is congested – it is advantages to send a small size packet first, this will keep the connection alive and have the least impact on bandwidth resources. Another

motivation is a fast start – by sending out a small size packet first, the receiver would receive some data faster and would start the communication earlier.

14. Claims 6-7, 14-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Applicant admitted prior art, further in view of McCullough as applied to claims 1, 9 above, and further in view of Kitamura.

Claim 6. AAPA does not teach the combination of these limitations: *generating the first outgoing sub-header comprises modifying the MF, Offset, Length, and Checksum fields of the incoming IP header according to the predetermined portion of the payload stored in the first storage unit*.

In the same field of endeavor, Kitamura discloses *generating the first outgoing sub-header comprises modifying the MF, Offset, Length, and Checksum fields of the incoming IP header according to the predetermined portion of the payload stored in the first storage unit* (Fig. 8, p6 [0118] expanded PPPoE portion, an upper layer identifier, a version, a type, a TOS, a data length, an identifier, a flag, a fragment offset, a TTL, an upper layer identifier, a header checksum, a source IP address, a destination IP address are written).

It would have been obvious to a person having ordinary skill in the art, at the time that the invention was made, to combine the teachings of Kitamura with AAPA to follow the same arrangement as is customary in packet communications when one packet is fragmented into multiple packets.

Claim 7. AAPA does not teach the combination of these limitations: *generating the second outgoing sub-header comprises modifying the MF, Offset, Length, and Checksum fields of the incoming packet IP header or the first outgoing sub-header according to the remaining portion of the payload stored in the storage units*.

In the same field of endeavor, Kitamura discloses *generating the second outgoing sub-header comprises modifying the MF, Offset, Length, and Checksum fields of the incoming packet IP header or the first outgoing sub-header according to the remaining portion of the payload stored in the storage units* (Fig. 8, [0118] expanded PPPoE portion, an upper layer identifier, a version, a type, a TOS, a data length, an identifier, a flag, a fragment offset, a TTL, an upper layer identifier, a header checksum, a source IP address, a destination IP address are written).

It would have been obvious to a person having ordinary skill in the art, at the time that the invention was made, to combine the teachings of Kitamura with AAPA to follow the same arrangement as is customary in packet communications when one packet is fragmented into multiple packets.

Claim 14. AAPA does not teach the combination of these limitations: *generating the first outgoing sub-header modifies the MF, Offset, Length, and checksum fields of the incoming packet IP header according to the first fragment*.

In the same field of endeavor, Kitamura discloses *generating the first outgoing sub-header modifies the MF, Offset, Length, and checksum fields of the incoming*

packet IP header according to the first fragment (Fig. 8, [0118] expanded PPPoE portion, an upper layer identifier, a version, a type, a TOS, a data length, an identifier, a flag, a fragment offset, a TTL, an upper layer identifier, a header checksum, a source IP address, a destination IP address are written).

It would have been obvious to a person having ordinary skill in the art, at the time that the invention was made, to combine the teachings of Kitamura with AAPA to follow the same arrangement as is customary in packet communications when one packet is fragmented into multiple packets.

Claim 15. AAPA does not teach the combination of these limitations: *generating the second outgoing sub-header modifies the MF, offset, length, and checksum fields of the incoming packet IP header or the first outgoing sub-header according to the second fragment*.

In the same field of endeavor, Kitamura discloses *generating the second outgoing sub-header modifies the MF, offset, length, and checksum fields of the incoming packet IP header or the first outgoing sub-header according to the second fragment* (Fig. 8, [0118] expanded PPPoE portion, an upper layer identifier, a version, a type, a TOS, a data length, an identifier, a flag, a fragment offset, a TTL, an upper layer identifier, a header checksum, a source IP address, a destination IP address are written).

It would have been obvious to a person having ordinary skill in the art, at the time that the invention was made, to combine the teachings of Kitamura with AAPA to follow

the same arrangement as is customary in packet communications when one packet is fragmented into multiple packets.

Response to Arguments

15. Applicant's arguments filed on 03/20/2009, regarding utilization of design choice on pages 1 - 4, with respect to amended claims have been considered but are moot in view of the new ground(s) of rejection. A new reference has been provided that meets the new limitation.

16. A second argument, starting on page 4 second paragraph to the end of page 4, is that claim 1 lines 1-10 is disclosed by the Applicant Admitted Prior Art, and only the remaining limitation lines 11-12 is not disclosed by the AAPA; the primary reference should be the AAPA; the motivation to combine the primary reference and the AAPA is an improper conclusory statement that embodies clear and improper hindsight rationale.

17. As applicant has stated, the limitation, claim 1 lines 1-10, is well known in the art and many references are available that teach this limitation as detailed by the explanation provided by the AAPA. Both the primary reference utilized in the office action and the AAPA are in the field of IP data encapsulation in accordance with PPPoE protocol, they are both solving the same problem. As disclosed by the primary reference and the AAPA, fragment processing is required. It would have been obvious to a person having ordinary skill in the art, at the time that the invention was made, that the disclosure that fragment processing is required teaches that the payload of the incoming packet is fragmented into portions for re-transmission. Furthermore, it would

have been obvious that storage is needed to store the incoming packet and memory management is needed for fragment processing. Nevertheless, the primary reference does not explicitly teach the combination of these features. The AAPA explicitly teaches this feature. Since both of these references are in the same field of endeavor and they are both solving the same problem, they are combinable. One motivation to combine is to provide the explicit hardware implementation of the teachings of the primary reference.

18. The primary reference and the AAPA had been combined in the first office action. It is unclear why in the response to the first office action, response dated 07/31/2008, the above issues had not been brought up. Furthermore, there had been a conflict between the drawings and the specification regarding the AAPA which had been corrected in the form of a replacement drawing of Fig. 3 Prior art, which was filed in response to the first office action.

19. Examiner respectfully points out that in the instant communication, the applicant has admitted that claim 1 lines 1-10 is disclosed by the Applicant Admitted Prior Art – which is the same as claim 1 as originally filed.

20. Furthermore, page 4 second paragraph lines 3-4 state "*the Office Action combined Ko with Miyahara to reject the claims*". Examiner respectfully points out that these references had not been utilized in the office action.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Hooman Houshmand whose telephone number is (571) 270-1817. The examiner can normally be reached on Monday - Friday 8am - 5pm EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Jay Patel can be reached on (571) 272-2988. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/H. H./
Examiner, Art Unit 2419

/Jayanti K. Patel/

Supervisory Patent Examiner, Art Unit 2419